Reading Week Exercises

Data Acquisition and File Exception Handling

Python File and Exception Handling

1. Write a Python program to open a file named "example.txt" in write mode, write "Hello, World!" to it, and then close the file.
2. Modify the program in question 1 to handle the exception if the file cannot be opened.
3. Write a Python program to read the contents of "example.txt" and print them to the console. Use exception handling to catch any errors.
4. Write a Python program that attempts to open a non-existent file "missing.txt" and handles the FileNotFoundError gracefully.
5. Create a Python program that writes a list of numbers (1 to 10) to a file, then reads the file and prints each number. Handle any potential exceptions.
6. Write a Python program that opens a file, reads its content line by line using a loop, and prints each line. Include exception handling.
7. Write a Python program that attempts to divide two numbers, taking user input, and handles the ZeroDivisionError if the user enters zero as the denominator.
8. Modify question 7 to handle ValueError if the user inputs a non-numeric value instead of a number.
9. Write a Python program that attempts to open a file and uses a try-except-finally block to ensure the file is always closed properly.
10. Write a Python program that checks if a file exists before opening it. If the file exists, print its contents; otherwise, print a message stating that the file does not exist.

Data Acquisition

11. Write a Python program to read data from a CSV file named "data.csv" and print its contents.
12. Write a Python program that reads an Excel file (data.xlsx) and prints the first five rows using the pandas library.
13. Write a Python script to collect stock market data from an API (e.g., Alpha Vantage or Yahoo Finance) and display the latest stock price for a given company.
14. Write a Python program to check if a given dataset file (dataset.csv) exists before attempting to read it. If it doesn't exist, print a message informing the user.

Web Scraping Questions

15. Write a Python script using requests and BeautifulSoup to extract the title of a webpage (e.g., "https://dbs.ie/").
16. Write a Python program that scrapes all the links (<a> tags) from the webpage in question 16 above and prints them.
17. Write a Python script that extracts and prints all headings (h1, h2, and h3 tags) from the given webpage in question 16 above.
18. Write a Python script that scrapes a table from the webpage in question 16 above and stores the data in a CSV file.